Software Support For Programming-in-the-Large

Final Report

1997

BAA: 90-21
Topic: Advanced Environments
ARPA Order: 8856
ONR Contract: N00014-92-J-1937
Period: September 15, 1992 to June 30, 1997
Total Support: $1,054,008

Thomas W. Reps, Principal Investigator
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706

1. Summary of the Completed Project

The goal of this project was to create enhanced tools to support the development of complex software systems. The objective was to create tools that provide powerful language-specific program-manipulation operations for aiding programmers in maintaining, enhancing, and reusing code. The main activities of the project were devoted to:

Program slicing

A particularly important aspect of the project was to explore how "program slicing" could serve as the basis for such program-manipulation operations. The slice of a program with respect to a set of program elements *S* is a projection of the program that includes only program elements that might affect (either directly or transitively) the values of the variables used at members of *S*. Slicing allows one to find semantically meaningful decompositions of programs—where the decompositions consist of elements that are not textually contiguous. Program slicing is a fundamental operation that can aid in solving many software-engineering problems. For instance, it has applications to program understanding, maintenance, debugging, testing, differencing, specialization, reuse, and merging.

The first phase of the work (1988–94, which was funded by a previous ARPA contract—A.O. 6378—as well as the first two years of A.O. 8856) focused on developing the theoretical foundations of program slicing. The later phases of the project focused on the construction and evaluation of a slicing tool for C, which has been used to investigate the practicality of slicing. Activities included:

- Improving the underlying technology for program slicing (and related operations).
- Implementing program slicers.
- Developing methods for using slicing in software-engineering tools.
- Building slicing-based program-manipulation tools.

This work was reported in publications [8], [13], [14], [15], [16], [23], [26], [27], [37], [38], [39], [40], [41], [43], [46], [47], and [54].

A new approach to program analysis

We discovered that a number of different program-analysis problems could be solved by transforming them to graph-reachability problems. Some of the program-analysis problems that are amenable to this treatment include program slicing, certain dataflow-analysis problems, and the problem of approximating the possible "shapes" that heap-allocated structures in a program can take on. Relationships between graph reachability and other approaches to program analysis were explored. Some techniques that go beyond pure graph reachability were also developed. This work was reported in publications [4], [5], [10], [12], [20], [25], [28], [29], [30], [32], [33], [48], and [52].

Techniques for pointer analysis and shape analysis

Techniques for program manipulation require knowledge of what items pointer variables point to, as well as a characterization of the "shapes" of the structures that pointer variables point to. Our work on this subject was reported in publications [2], [3], [19], [21], [24], [49], and [50].

Incremental computation

Motivated by the goal of speeding up response times of interactive systems, we explored techniques for dealing with "incremental changes" to the inputs of computations. This work was reported in publications [1], [6], [7], [9], [34], [55], and [56].

Development of the CAPITL program-development environment

CAPITL consists of:

- A shared, object-oriented, versioned database.
- An embedded logic-based data-manipulation language.
- A graphical user interface.

With each software object the database stores a rich set of attributes that describe its syntax, intended semantics, and relationship to other objects. CAPITL is implemented in POL, a data model and deductive query language with elements of persistent, object-oriented and logic-based programming languages. POL is implemented in and tightly coupled with C++. This work was reported in publications [31], [36], [44], and [53].

Publications [10], [32], and [47] are included with this report.

2. Publications Written Under ARPA Order 8856 (ONR Contract N00014-92-J-1937)

The following papers were written by the Principal Investigator, the co-Investigators, and their students during the period of ARPA Order 8856 (ONR Contract N00014-92-J-1937). Most of the papers are available over the World Wide Web at either http://www.cs.wisc.edu/~reps/ or http://www.cs.wisc.edu/wpis/html.

Books

[1] Ramalingam, G., *Bounded Incremental Computation*, Lecture Notes in Computer Science, Vol. 1089, Springer-Verlag, New York, NY, 1996.

Journal Publications

[2] Sagiv, M., Reps, T. and Wilhelm, R., Solving shape-analysis problems in languages with destructive updating. To appear in *ACM Transactions on Programming Languages and Systems.*

[3] S. Horwitz, Precise flow-insensitive may-alias analysis is NP-hard, *ACM Transactions on Programming Languages and Systems 19*, 1 (January 1997).

[4] Sagiv, M., Reps, T., and Horwitz, S., Precise interprocedural dataflow analysis with applications to constant propagation. *Theoretical Computer Science 167* (1996), pp. 131-170. (Special issue of papers from TAPSOFT '95.)

[5] Reps, T., On the sequential nature of interprocedural program-analysis problems. *Acta Informatica 33* (1996), pp. 739-757.

[6] Ramalingam, G. and Reps, T., An incremental algorithm for a generalization of the shortest-path problem. *Journal of Algorithms 21* (1996), pp. 267-305.

[7] Ramalingam, G. and Reps, T., On the computational complexity of dynamic graph problems. *Theoretical Computer Science A 158* (May 1996), pp. 233-277.

[8] Binkley, D., Horwitz, S., and Reps, T., Program integration for languages with procedure calls. *ACM Transactions on Software Engineering and Methodology 4*, 1 (January 1995), pp. 3-35.

[9] Ramalingam, G. and Reps, T., On competitive on-line algorithms for the dynamic priority-ordering problem. *Information Processing Letters 51* (1994), 155-161.

Invited Papers

[10] Reps, T., Program analysis via graph reachability. In *Proc. of ILPS '97: International Logic Programming Symposium*, (Port Jefferson, NY, Oct. 12-17, 1997), J. Maluszynski (ed.), The M.I.T. Press, Cambridge, MA, 1997, pp. 5-19.

[11] Reps, T., The use of program profiling in software testing. In *Proc. of Informatik '97* (Aachen, Germany, Sept. 24-27, 1997), M. Jarke, K. Pasedach, and K. Pohl (eds.), Springer-Verlag, Berlin, Ger., 1997, pp. 4-16.

Book Chapters

[12] Reps, T., Demand interprocedural program analysis using logic databases, In *Applications of Logic Databases*, R. Ramakrishnan (ed.), Kluwer Academic Publishers, Boston, MA, 1994, pp. 163-196.

[13] Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. To appear in *Software Change Impact Analysis*, S.A. Bohner and R.S. Arnold (eds.), IEEE Computer Society, Los Alamitos, CA.

Reprinted from *ACM Transactions on Programming Languages and Systems 12*, 1 (January 1990), 26-60.

[14] Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. In *Software Merging and Slicing*, V. Berzins (ed.), IEEE Computer Society, Los Alamitos, CA, pp. 10-44.

Reprinted from *ACM Transactions on Programming Languages and Systems 12*, 1 (January 1990), 26-60.

[15] Horwitz, S., Prins, J., and Reps, T., Integrating non-interfering versions of programs. In *Software Merging and Slicing*, V. Berzins (ed.), IEEE Computer Society, Los Alamitos, CA, pp. 137-179.

Reprinted from *ACM Transactions on Programming Languages and Systems 11*, 3 (July 1989), 345-387.

[16] Ramalingam, G. and Reps, T., A theory of program modifications. In *Software Merging and Slicing*, V. Berzins (ed.), IEEE Computer Society, Los Alamitos, CA, 90-105.

Reprinted from *Proceedings of the Colloquium on Combining Paradigms for Software Development, Lecture Notes in Computer Science*, Vol. 494, S. Abramsky and T.S.E. Maibaum (eds.), Springer-Verlag, New York, NY, 1991, pp. 137-152.

Conference Publications

[17] Siff, M. and Reps, T., Identifying modules via concept analysis. In *ICSM '97: IEEE International Conference on Software Maintenance* (Oct. 1-3, 1997, Bari, Italy), M.J. Harrold and G. Visaggio (eds.), IEEE Computer Society, Washington, DC, 1997, pp. 170-179.

[18] Reps, T., Ball, T., Das, M., and Larus, J., The use of program profiling for software maintenance with applications to the Year 2000 Problem. In *Proceedings of ESEC/FSE '97: Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Zurich, Switzerland, Sept. 22-25, 1997), *Lecture Notes in Computer Science*, Vol. 1301, M. Jazayeri and H. Schauer (eds.), Springer-Verlag, New York, NY, 1997, pp. 432-449.

[19] Shapiro, M. and Horwitz, S., The effects of the precision of pointer analysis. In *Proceedings of the 4th International Symposium on Static Analysis*, (Paris, France, September 8-10, 1997), *Lecture Notes in Computer Science*, Vol. 1302, P. Van Hentenryck (ed.), Springer-Verlag, New York, NY, 1997, pp. 16-34.

[20] Melski, D. and Reps, T., Interconvertibility of set constraints and context-free language reachability. In *PEPM '97: Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, (Amsterdam, The Netherlands, June 12-13, 1997), ACM, New York, NY, 1997, pp. 74-89.

[21] Shapiro, M. and Horwitz, S., Fast and accurate flow-insensitive points-to analysis. In *Conference Record of the Twenty-Fourth ACM Symposium on Principles of Programming Languages*, (Paris, France, Jan. 15-17, 1997), ACM, New York, NY, 1997.

[22] Siff, M. and Reps, T., Program generalization for software reuse: From C to C++. In *SIGSOFT 96: Proceedings of the Fourth ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (San Francisco, CA, October 16-18, 1996), ACM, New York, NY, 1996, pp. 135-146.

[23] Reps, T. and Turnidge, T., Program specialization via program slicing. In *Proceedings of the Dagstuhl Seminar on Partial Evaluation*, (Schloss Dagstuhl, Wadern, Germany, Feb. 12-16, 1996), *Lecture Notes in Computer Science*, Vol. 1110, O. Danvy, R. Glueck, and P. Thiemann (eds.), Springer-Verlag, New York, NY, 1996, pp. 409-429.

[24] Sagiv, M., Reps, T., and Wilhelm, R., Solving shape-analysis problems in languages with destructive updating. In *Conference Record of the Twenty-Third ACM Symposium on Principles of Programming Languages*, (St. Petersburg, FL, Jan. 22-24, 1996), ACM, New York, NY, 1996, pp. 16-31.

[25] Horwitz, S., Reps, T., and Sagiv, M., Demand interprocedural dataflow analysis. In *SIGSOFT '95: Proceedings of the Third ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Washington, DC, October 10-13, 1995).

[26] Reps, T. and Rosay, G., Precise interprocedural chopping. In *SIGSOFT '95: Proceedings of the Third ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Washington, DC, October 10-13, 1995).

[27] Das, M., Reps, T., and Van Hentenryck, P. Semantic foundations of binding-time analysis for imperative programs. In *PEPM '95: Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, (La Jolla, California, June 21-23, 1995), ACM, New York, NY, 1995, pp. 100-110.

[28] Reps, T., Shape analysis as a generalized path problem. In *PEPM '95: Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, (La Jolla, California, June 21-23, 1995), ACM, New York, NY, 1995, pp. 1-11.

[29] Sagiv, M., Reps, T., and Horwitz, S., Precise interprocedural dataflow analysis with applications to constant propagation. In *Proceedings of FASE '95: Colloquium on Formal Approaches in Software Engineering*, (Aarhus, Denmark, May 22-26, 1995), *Lecture Notes in Computer Science*, Vol. 915, P.D. Mosses, M. Nielsen, and M.I. Schwartzbach (eds.), Springer-Verlag, New York, NY, 1995, pp. 651-665.

[30] Reps, T., Horwitz, S., and Sagiv, M., Precise interprocedural dataflow analysis via graph reachability. In *Conference Record of the Twenty-Second ACM Symposium on Principles of Programming Languages*, (San Francisco, CA, Jan. 23-25, 1995), ACM, New York, NY, 1995, pp. 49-61.

[31] Adams, P. and Solomon, M., An overview of the CAPITL software development environment, *Software Configuration Management, Lecture Notes in Computer Science*, Vol. 1005, Springer-Verlag, New York, NY, 1995.

[32] Reps, T., Horwitz, S., Sagiv, M., and Rosay, G. Speeding up slicing. In *SIGSOFT '94: Proceedings of the Second ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (New Orleans, LA, December 7-9, 1994), *ACM SIGSOFT Software Engineering Notes 19*, 5 (December 1994), pp. 11-20.

[33] Reps, T. Solving demand versions of interprocedural analysis problems. In *Proceedings of the Fifth International Conference on Compiler Construction*, (Edinburgh, Scotland, April 7-9, 1994), *Lecture Notes in Computer Science*, Vol. 786, P. Fritzson (ed.), Springer-Verlag, New York, NY, 1994, pp. 389-403.

[34] Ramalingam, G. and Reps, T. An incremental algorithm for maintaining the dominator tree of a reducible flowgraph. In *Conference Record of the Twenty-First ACM Symposium on Principles of Programming Languages*, (Portland, OR, Jan. 16-19, 1994), ACM, New York, NY, 1994, pp. 287-296.

[35] Reps, T., Scan grammars: Parallel attribute evaluation via data-parallelism. In *Proceedings of the Fifth ACM Symposium on Parallel Algorithms and Architectures*, (Velen, Germany, June 30 - July 2, 1993).

[36] Adams, P. and Solomon, M., An overview of the CAPITL software development environment. Fourth International Workshop on Software Configuration Management, Baltimore, MD, 1993.

[37] Ball, T. and Horwitz, S., Slicing programs with arbitrary control flow. In *Proceedings of the First International Workshop on Automated and Algorithmic Debugging*, (Linkoeping, Sweden, May 1993), *Lecture Notes in Computer Science*, Vol. 749, Springer-Verlag, New York, NY, 1993, pp. 206-222.

[38] Bates, S. and Horwitz, S., Incremental program testing using program dependence graphs. In *Conference Record of the Twentieth ACM Symposium on Principles of Programming Languages*, (Charleston, SC, January 10-13, 1993), ACM, New York, NY, 1993, pp. 384-396.

Software

[39] Reps, T., Rosay, G., and Horwitz, S., *The Wisconsin Program-Slicing Tool.* Release 1.0, August 1997.

[40] Reps, T., Bricker, T., Rosay, G., *et al., The Wisconsin Program-Integration System.* Release 0.5, April 1990; Release 1.0, April 1992. Release 2.0, July 1993. Licensed to 9 sites.

Patents

[41] Reps, T., Horwitz, S., and Binkley, D., Interprocedural slicing of computer programs using dependence graphs. U.S. Patent Number 5,161,216, November 3, 1992.

Pending Submissions

[42] Reps, T., "Maximal-munch" tokenization in linear time. TR-1347, Computer Sciences Department, University of Wisconsin, Madison, WI, May 1997.
Submitted for journal publication.

[43] Ramalingam, G. and Reps, T., New programs from old. Computer Sciences Department, University of Wisconsin–Madison.

Submitted for journal publication.

Ph.D. Dissertations

[44] Adams, P., A logical framework for software construction. Ph.D. dissertation, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1994.

[45] Ramalingam, G., Bounded incremental computation. Ph.D. dissertation and Tech. Rep. TR-1172, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1993.

[46] Ball, T.J., The use of control-flow and control dependence in software tools. Ph.D. dissertation and Tech. Rep. TR-1169, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1993.

Other Publications and Reports

[47] *The Wisconsin Program-Slicing Tool 1.0, Reference Manual.* Computer Sciences Department, University of Wisconsin-Madison, August 1997.

[48] Das, M. and Reps, T. BTA termination using CFL-reachability. TR-1329, Computer Sciences Department, University of Wisconsin, Madison, WI, November 1996.

[49] Reps, T., Sagiv, M., and Wilhelm, R., On shape abstractions and shape analyses. Computer Sciences Department, University of Wisconsin, Madison, WI, July 1996.

[50] Sagiv, M., Reps, T. and Wilhelm, R., Solving shape-analysis problems in languages with destructive updating. TR-1276, Computer Sciences Department, University of Wisconsin–Madison, July 1995.

[51] van Leeuwen, J., Mehlhorn, K., and Reps, T. (eds.), Incremental computation and dynamic algorithms. Dagstuhl Seminar Report 88, International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 1994.

[52] Reps, T., Sagiv, M., and Horwitz S., Interprocedural dataflow analysis via graph reachability. TR 94-14, Datalogisk Institut, University of Copenhagen, Copenhagen, Denmark, April 1994.

[53] Adams, P. and Solomon, M., POL: Persistent objects with logic, TR-1158, Computer Sciences Department, University of Wisconsin–Madison, August 1993.

[54] Reps, T., *The Wisconsin Program-Integration System Reference Manual: Release 2.0.* Computer Sciences Department, University of Wisconsin–Madison, July 1993.

[55] Ramalingam, G. and Reps, T., A categorized bibliography on incremental computation. In *Conference Record of the Twentieth ACM Symposium on Principles of Programming Languages*, (Charleston, SC, Jan. 11-13, 1993), ACM, New York, NY, 1993, pp. 502-510. (Tutorial paper.)

[56] Reps, T., Incremental computation. Unpublished tutorial notes, 1993. (Presented at the Twentieth ACM Symposium on Principles of Programming Languages, (Charleston, SC, Jan. 11-13, 1993).)

[57] Klint, P., Reps, T., and Snelting, G. (eds.), Programming environments. Dagstuhl Seminar Report 34, International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 1992.